

# Minimal Disclosure in Partially Observable Markov Decision Processes

Nathalie Bertrand<sup>1</sup> and Blaise Genest<sup>2</sup>

<sup>1</sup> INRIA Rennes Bretagne Atlantique, France

<sup>2</sup> CNRS, UMI IPAL, joint with NUS and A\*STAR/I2R, Singapore

---

## Abstract

For security and efficiency reasons, most systems do not give the users a full access to their information. One key specification formalism for these systems are the so called *Partially Observable Markov Decision Processes* (POMDP for short), which have been extensively studied in several research communities, among which AI and model-checking. In this paper we tackle the problem of the *minimal information* a user needs *at runtime* to achieve a simple goal, modeled as reaching an objective with probability one. More precisely, to achieve her goal, the user can at each step either choose to use the partial information, or pay a fixed cost and receive the full information. The natural question is then to minimize the cost the user needs to fulfill her objective. This optimization question gives rise to two different problems, whether we consider to minimize the *worst case cost*, or the *average cost*. On the one hand, concerning the worst case cost, we show that efficient techniques from the model checking community can be adapted to compute the optimal worst case cost and give optimal strategies for the users. On the other hand, we show that the optimal average price (a question typically considered in the AI community) cannot be computed in general, nor can it be approximated in polynomial time even up to a large approximation factor.

## 1 Introduction

Partially Observable Markov Decision Processes (POMDP for short) form a powerful model to describe systems where part of the information is not accessible at runtime by the user, and where the effect of the user actions is randomized. Partial observation happens virtually in every real life systems for various reasons, *e.g.* complexity, privacy or security. The usual question is given the observation at runtime and the (offline) POMDP description of the complete system, can a user achieve some goal or optimize some value?

In this paper, rather than considering that the partial information is rigidly fixed, we aim at evaluating several observation schemes. In applications where partial observation arises from complexity reasons, the system should provide *at runtime* the weakest observation which still allows to achieve a given goal, as it would also have the minimum cost of deployment. On the contrary, in the context of security, the objective is to design a secure system preventing an attacker to achieve her goal, by giving her only partial access to the state of the system. Knowing that the partial observation scheme may be vulnerable, one may be extremely careful and analyze the additional information (obtained by punctually attacking the partial observation) an attacker needs *at runtime* to achieve her objective.

We analyze in depth the simplest such framework to which many problems can be reduced: The aim of the user is to reach with probability 1 a set **Goal** of states. We consider only two alternatives of information: by default, the user gets a fixed partial information; If requested, she can also obtain full information on the current state of the system. In this framework, each execution is naturally assigned a “cost” – the number of times the full information is requested – and the user always aims at minimizing this amount. Now, *giving*



a *value* to a strategy reaching **Goal** can be done in two different meaningful ways: either the *worst case cost* the user can have to pay or the *average cost* she pays, while following this strategy. For both options, the objective is to compute the optimal cost (among almost-sure winning strategies) and if possible synthesize a family of strategies approximating this optimal for smaller and smaller approximation factors.

A first contribution is to show that efficient model-checking techniques can be adapted to compute the worst case cost. Furthermore, we design optimal strategies in such cases and prove that strategies with finite memory, based on the set  $\mathcal{B}$  of so-called (discrete) belief states, are optimal. A belief state represents the set of states the system can be in after a given sequence of actions and observations. First, we check in polynomial time that **Goal** can be reached with probability 1 using **req** unrestrictedly. If it is not the case, the whole procedure is pointless. Assuming **Goal** can be reached almost surely, we define a family of generic strategies  $(\sigma_{can}^n)_{n \in \mathbb{N}}$  with memory state in  $\mathcal{B}$  and which is almost-surely winning. To improve the worst case cost of these strategies, we compute the set of states from which **Goal** can be reached with probability 1 without ever requesting the full information, then the set of states requesting the full information at most once, at most twice etc. The whole process terminates in time polynomial in  $|\mathcal{B}|$ , and gives associated strategies with finite memory  $\mathcal{B}$ . Of course,  $|\mathcal{B}|$  is at worst exponential in the number of states of the POMDP. The memory-size of these strategies, as well as the complexity of our algorithm are thus optimal since deciding whether **Goal** can be reached almost-surely without any requests (a simple subproblem) is EXPTIME-complete and requires strategies with exponential memory [6]. We illustrate the approach on a simple example for which we analyze the family of strategies, and show that it is optimal also for average cost on this particular instance.

However, this optimality result for average cost in this simple example is far from being generally true. Indeed, first of all, we prove that computing optimal average cost for a POMDP under a reachability objective is undecidable. Even worse, it is undecidable to even approximate it, whatever the approximation factor. At last, we give non approximability factors exponential in the size of the system, and prove that there is no algorithm running in polynomial time in the number of belief states to approximate the optimal average cost within that factor, even over finite horizon, for which the undecidability result does not hold.

## Related work

The POMDP model has been studied by at least two communities: first by the Artificial Intelligence and Operations Research community where mostly the problem consists in optimizing a reward function. Here, the results are twofold. First, they propose heuristics to obtain policies to get good rewards, using value iteration, grid-based algorithms [11], strategy improvement [10], etc.; see for instance [14] for a survey. Also, they analyze the complexity class in which such problem falls: It is undecidable in general to compute or approximate the optimal reward [13], and it is NP-complete to do it in finite horizon [12]. Compared to these works, we consider a particular cost function which cannot be expressed as a reward, and our lower bound results needs only a polynomial number of belief states.

More recently, the Model Checking community considered POMDP, where the question was mainly qualitative (probability 1 or positive) over a navigational goal: reachability or safety (avoiding to reach a state), once or repeatedly. The problem of reachability with probability 1 is the dual problem of safety with positive probability. These problems are EXPTIME-complete in general [6] and PSPACE-complete when the user has no observation at all [16, 4]. While visiting infinitely often a state almost-surely has the same complexity as almost sure reachability, reaching infinitely often a state with positive probability is unde-



cidable for POMDP [1]. Surprisingly, with a slight constraint on these infinitely many visits, namely that the limit average number of times the goal is visited shall be positive, this last problem is decidable [16]. More complex systems than POMDP have been considered, where two partially informed players have opposite objectives: the results on POMDP mentioned above basically carry over [2]. More general winning conditions have also been studied [5]. Only very recently (up to our knowledge), both a particular numerical function (energy) and navigational goal were considered [9].

The closest work to ours is by Chatterjee et al [8, 7], where the problem is the existence of a controller with limited budget (and of an optimal controller) achieving an omega-regular objective in an imperfect information game. In this work and similarly to ours, partial observation of the controller is not fixed and varies according to the choices of the controller to allocate its budget. The main difference from our setting is that the arena as well as the strategy of the controller are *pure*, that is, they do not incorporate probabilities.

Controllers that dynamically request access to more precise information (e.g. by activating sensors) has also been studied in control and diagnosis for discrete event systems [15, 3].

## 2 Notations

Given  $S$  a finite set, let  $\text{Dist}(S)$  denote the set of *distributions* over  $S$ , that is functions  $d : S \rightarrow [0, 1]$  such that  $\sum_{s \in S} d(s) = 1$ . The *support* of distribution  $d$  is defined as  $\text{Supp}(d) = \{s \in S \mid d(s) > 0\}$ . If  $d$  is the *Dirac distribution* associated with  $s \in S$  (that is  $d(s) = 1$  and  $d(t) = 0$  for every  $t \neq s$ ) we will abuse notation and simply write  $d = s$ .

► **Definition 1** ((PO)MDP). A *Markov decision process* (MDP) is tuple  $(Q, \text{Act}, \Delta)$  where  $Q$  is a finite set of states,  $\text{Act}$  is a finite set of actions, and  $\Delta : Q \times \text{Act} \rightarrow \text{Dist}(Q)$  is the transition function.

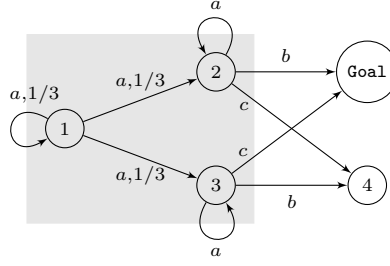
A *partially observable MDP* (POMDP) is a tuple  $\mathcal{M} = (Q, \text{Act}, \Delta, \text{Part})$ , where  $(Q, \text{Act}, \Delta)$  is an MDP, and  $\text{Part}$  is a partition of  $Q$ . Elements of  $\text{Part}$  are called *observations*.

Given a POMDP  $(Q, \text{Act}, \Delta, \text{Part})$  the underlying MDP  $(Q, \text{Act}, \Delta)$  is alternatively called the FOMDP, for *Fully Observable* MDP. Intuitively, in a POMDP, from state  $s$ , if action  $a$  is chosen, the next state is  $t$  with probability  $\Delta(s, a)(t)$  and the controller receives observation  $O \in \text{Part}$  such that  $p \in O$ . Of course, any MDP can be seen as a POMDP by setting  $\text{Part}$  as the set of all singletons sets. In the sequel, we always assume a fixed starting state  $s_0$  and also distinguish a goal (set of) state(s)  $\text{Goal}$ . We assume for convenience that being in  $\text{Goal}$  is observable, that is for all  $P \in \text{Part}$ ,  $P \subseteq 2^{\text{Goal}} \cap P$  or  $P \subseteq 2^{Q \setminus \text{Goal}}$ .

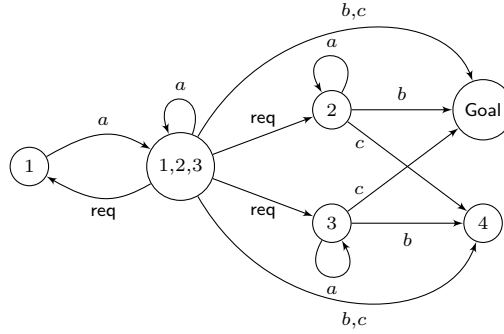
An example of POMDP is given in Figure 1, where  $\text{Part} = \{\{1, 2, 3\}, \{\text{Goal}\}, \{4\}\}$  and the non-trivial set of the partition is represented by a grey area.

In this paper, given a POMDP  $(Q, \text{Act}, \Delta, \text{Part})$ , we assume that the controller can perform an extra action  $\text{req} \notin \text{Act}$  whose effect is to disclose the precise state of the FOMDP. More precisely, the observation the controller receives after a request action  $\text{req}$  is  $\{s\}$  if the current state is  $s \in Q$ . The set of possible observations, denoted  $\mathcal{O}$ , thus consists of the partition  $\text{Part}$ , together with  $\text{Single} = \{\{s\} \mid s \in Q\}$ . The probabilistic transition function is extended to  $\text{Act}' = \text{Act} \cup \{\text{req}\}$  by defining  $\Delta(s, \text{req})$  as the Dirac distribution associated with observation  $\{s\}$ . Our high-level aim is to design a strategy of choosing actions to play based only on the sequence of actions played and observations received, such that the goal can be reached. Notice that from  $s_0 = 1$ , one needs to play  $a$  as no other choice is possible. A possible outcome is to reach state 1 again, hence for any choice of actions, there will always be a possibility to not reach  $\text{Goal}$ . Hence our aim is instead to ensure that  $\text{Goal}$  is





■ **Figure 1** A simple POMDP



■ **Figure 2** Graph of reachable belief states for the POMDP in Figure 1

reached, unless being terribly unlucky. Formally, we want the set of paths reaching **Goal** to have probability 1. We define the probability space for  $G$  by extending the POMDP to  $\text{Act}'$ .

## 2.1 Belief States

Let  $(a_1, O_1) \cdots (a_i, O_i) \in (\text{Act} \times \text{Part} \cup \{\text{req}\} \times \text{Single})^*$  be a sequence of actions played and observations received. We define  $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$  as the set of states  $s \in Q$  such that for all  $0 \leq j \leq i$  there exists  $t_j$  with  $t_0 = s_0$ ,  $t_i = s$ ,  $t_j \in O_j$  and  $\Delta(t_{j-1}, a_j)(t_j) > 0$ . Intuitively if actions  $a_1 \cdots a_i$  have been played and observations  $O_1 \cdots O_i$  were received, the set of possible states are exactly those in  $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$ . The set  $B(s_0, (a_1, O_1) \cdots (a_i, O_i))$  can be computed inductively:

$$B(s_0, (a_1, O_1) \cdots (a_i, O_i)) = O_i \cap \{t \mid \exists s \in B(s_0, (a_1, O_1) \cdots (a_{i-1}, O_{i-1})), \Delta(s, a_i)(t) > 0\}.$$

The set **Path** of *possible finite paths* consists in all  $\rho = (a_1, O_1) \cdots (a_n, O_n) \in (\text{Act} \times \text{Part} \cup \{\text{req}\} \times \text{Single})^*$  such that  $B(s_0, (a_1, O_1) \cdots (a_n, O_n)) \neq \emptyset$ . Moreover, the set of *reachable belief states* is defined by  $\mathcal{B} = \{B(s_0, \rho) \mid \rho \in \text{Path}\}$ . For the example POMDP of Figure 1 the graph of reachable belief states is depicted in Figure 2. The number of belief states is at worst exponential in the number of states, but often, it is not that big.

Given a finite path  $\rho$ ,  $B(s_0, \rho)$  only gives the set of states the POMDP can be after executing  $\rho$ , but the precise probability to be in each state is unknown. The distributions  $D(s_0, \rho)$  and  $D(s_0, \rho, a)$  over states after a prefix of path (that is,  $\rho$  or  $\rho a$ , with  $\rho \in \text{Path}$  and  $a \in \text{Act}'$ ) can be defined by induction. Assuming  $\rho = \rho'(a, O)$ , and  $D(s_0, \rho')$  is known,  $D(s_0, \rho', a)$  is defined by  $D(s_0, \rho', a)(s) = \sum_t D(s_0, \rho')(t) \times \Delta(t, a)(s)$ . Then,



taking the observation into account yields:  $D(s_0, \rho)(s) = 0$  if  $s \notin O$ , else  $D(s_0, \rho)(s) = D(s_0, \rho', a)(s) / \sum_{s \in O} D(s_0, \rho', a)(s)$ .

Of course, the discrete belief after  $\rho$  can be recovered from the probabilistic belief:  $B(s_0, \rho) = \text{Supp}(D(s_0, \rho))$ .

## 2.2 Strategy

A *strategy*  $\sigma : \text{Path} \rightarrow \text{Dist}(\text{Act}')$  for the controller is a function which associates with any possible path a distribution over the extended set of actions  $\text{Act}'$ . Given a strategy  $\sigma$ , a possible path  $(a_1, O_1) \cdots (a_n, O_n) \in \text{Path}$  is called a  $\sigma$ -*path* if  $a_{i+1} \in \text{Supp}(\sigma((a_1, O_1) \cdots (a_i, O_i)))$  for all  $i < n$ . When a fixed strategy  $\sigma$  is played, it is easy to define the probability that a possible path occurs, as follows. First of all, if  $\rho = \rho'(a, O)$  is a  $\sigma$ -path,  $\mathbb{P}_\sigma(\rho) = \mathbb{P}_\sigma(\rho') \cdot \sigma(\rho')(a) \cdot \sum_{s \in O} D(\rho', a)(s)$ , and  $\mathbb{P}(\rho) = 1$  if  $\rho$  is the empty path. Now, we can define the probability of a run  $r = s_0 \xrightarrow{a_1} s_1 \cdots \xrightarrow{a_n} s_n$  (every  $s_i$  is a state of the FOMDP) knowing that  $\rho$  occurred. Let  $\rho$  be the unique (POMDP) path associated with  $r$ , obtained by replacing states with their associated observations. The probability of  $r$  assuming  $\rho$  is performed is given by  $\mathbb{P}(r \mid \rho) = \prod_{i \leq n} D(s_0, \rho_i)(s_i)$ , where  $\rho_i$  denotes the prefix of length  $i$  of  $\rho$ . Then we let  $\mathbb{P}_\sigma(r) = \mathbb{P}_\sigma(\rho) \times \mathbb{P}(r \mid \rho)$ . This probability measure on finite runs is extended in the usual way to the sigma-algebra they generate, and it is well-known that LTL properties on infinite runs are measurable for this measure [17]. The objective for the controller is to reach the **Goal** state. Thus, strategy  $\sigma$  is *almost-surely winning* if  $\mathbb{P}_\sigma(\Diamond \text{Goal}) = 1$ . Clearly enough, if there is no almost-surely winning strategy in the FOMDP, then there will be none in the extended POMDP either.

### Problem statement

To every strategy  $\sigma$ , two quantities can be associated: first of all, the worst-case cost, *i.e.* the maximum number of request actions  $\sigma$  takes along a  $\sigma$ -path; and second the average cost, that is the expected number of requests for  $\sigma$ -paths. In this paper we thus tackle the two distinct problems of finding almost-surely winning strategies which (1) minimize the worst-case cost, or (2) minimize the average cost.

## 3 Algorithms for the optimal worst case cost

Strategies are in general objects that do not have a finite presentation. In order to represent them effectively, it is common to restrict to finite-memory strategies, that are weaker than general strategies, but, as we will see, suffice when considering the optimal worst case cost problem. A *finite-memory strategy* on finite memory set  $M$  is given as  $\sigma : M \rightarrow \text{Dist}(\text{Act}')$  together with an update function:  $\text{up} : M \times (\text{Act} \times \text{Part} \cup \{\text{req}\} \times \text{Single}) \rightarrow M$ .

### 3.1 Reaching Goal with probability 1

We first propose a family  $(\sigma_{can}^n)_{n \in \mathbb{N}}$  of strategies with finite memory  $\mathcal{B} \subseteq 2^Q$  the set of reachable belief states. The memory is initialized to  $\{s_0\}$ , where  $s_0$  is the initial state. The update function is given by  $\text{up}(S, a, O) = O \cap \{t \mid \exists s \in S, \Delta(s, a_i)(t) > 0\}$ . We extend it inductively to a path  $\rho' = \rho \cdot (a, O)$  with  $\text{up}(S, \rho \cdot (a, O)) = \text{up}(\text{up}(S, \rho), a, O) = T$ , and we say that the path  $\rho'$  reaches the memory state  $T$ . It is easy to see that the memory state  $M$  reached after some possible path  $\rho$  is exactly  $B(s_0, \rho)$ .

Let  $n \in \mathbb{N}$ . We now define how the strategy  $\sigma_{can}^n$  plays. First, we denote by  $\text{Lose}_F \subseteq Q$  the set of states  $s$  of the FOMDP associated with  $G$  such that there is no strategy in the



FOMDP reaching **Goal** with probability 1 from  $s$ . We then denote by  $\text{Lose} \subseteq 2^Q$  the set of belief states  $S$  such that  $S \cap \text{Lose}_F \neq \emptyset$ . In the example of Figure 1, 4 is the only state of the FOMDP from which there is no strategy reaching **Goal** almost-surely. In the reachable belief graph of Figure 2,  $\text{Lose}$  is thus made of the single belief state  $\{4\}$ . Intuitively, any path reaching a memory state in  $\text{Lose}$  cannot reach **Goal** with probability 1. It is easy to see that if  $\{s_0\} \in \text{Lose}$ , then there is no almost-surely winning strategy, since otherwise a strategy in the FOMDP reproducing  $\sigma$  (which is possible since it has at least as much information) would also reach **Goal** with probability 1.

Letting  $\text{Win} = \mathcal{B} \setminus \text{Lose}$  the complementary set, we prove in the next theorem that under strategy  $\sigma_{can}^n$ , **Goal** is reached almost-surely from any belief state of  $\text{Win}$ . We partition  $\text{Win}$  in 3 sets:  $W_0 = \{S \mid S \subseteq \text{Goal}\}$  wins directly,  $W_N$  the *needing* belief states, and  $W_U$  the *non-needing* belief states. A state  $S$  is in  $W_N$  whenever for all  $a \in \text{Act}$ , there exists  $O \in \text{Part}$ , such that  $B(S, a, O) \in \text{Lose}$ . Intuitively, from a memory state in  $W_N$  any almost-surely winning strategy needs to perform a request action **req**, as every other action leaves a chance to reach  $\text{Lose}$ . In the example of Figure 2,  $W_0$  only contains the belief state  $\{\text{Goal}\}$ ,  $W_N = \emptyset$  (in particular, actions  $a$  can be done safely from belief states  $\{1\}$  and  $\{1, 2, 3\}$ ), and  $W_U$  consists of the other belief states  $\{1\}, \{1, 2, 3\}, \{2\}, \{3\}$ . Strategy  $\sigma_{can}^n$  is then defined by:

- If  $M \in W_0$ , then  $\sigma_{can}^n(M) = \emptyset$ ,
- if  $M \in W_N$ ,  $\sigma_{can}^n(M) = \text{req}$ ,
- if  $M \in W_U$ ,  $\sigma_{can}^n(M)$  plays **req** with probability  $1/n$ , and plays uniformly all actions  $a \in \text{Act}$  such that for every observation  $O$ ,  $\text{up}(M, a, O) \notin \text{Lose}$ , and
- if  $M \in \text{Lose}$ , then  $\sigma_{can}^n(M)$  is the uniform distributions over all actions.

Furthermore, if  $M \in \text{Single}$  (that is, the actual state is known for sure), then we disallow  $\sigma_{can}^n$  to perform a **req** as it would be useless. Notice that the only infinite  $\sigma_{can}^n$ -paths are exactly those which never meet  $W_0$ .

► **Theorem 2.** *If  $\{s_0\} \in \text{Win}$ , then  $\mathbb{P}_{\sigma_{can}^n}(\Diamond \text{Goal}) = 1$ .*

Notice that we can compute in polynomial time the set of losing state in the FOMDP, and hence decide in polynomial time whether there exists an almost-surely strategy in  $G$ .

### 3.2 Optimizing for the worst case cost

Now that we know the set of belief states from which there is a strategy reaching **Goal** with probability 1, we can tune the canonical strategies  $\sigma_{can}^n$ . To do so, we compute inductively the set  $S_k$  of belief states from which one needs at most  $k$  actions **req** to win. The set  $S_0$  is pretty easy to obtain, as the associated strategy cannot use any **req**.

Let  $G_0 = (\mathcal{B}, \text{Act}, \delta_0)$  be the *belief MDP* associated with the POMDP  $G$ , where each state is a belief state, and  $\delta_0(B, a)$  is the uniform distribution over all belief states  $B'$  such that there exists a part  $P \in \text{Part}$  with  $B' = P \cap \{t \mid \exists s \in B, \Delta(s, a)(t) > 0\}$ . Notice that requests are not allowed in  $G_0$ . The MDP  $G_0$  obtained from the POMDP in Figure 1 is very similar to the graph of Figure 2, except that the 3 edges labeled with **req** have been deleted. Let us denote by  $S_0$  the set of belief states from which there exists a strategy  $\sigma_0$  in  $G_0$  reaching  $W_0 = 2^{\text{Goal}}$  almost-surely. The set  $S_0$  can be computed in time linear in the number  $|\mathcal{B}|$  of states of  $G_0$ , thus at worst in time exponential in the size of  $G$ . Taking our example of Figure 1,  $S_0 = \{\{2\}, \{3\}, \{\text{Goal}\}\}$ . Indeed,  $\{4\}$  is a losing state, and  $\{1\}$  and  $\{1, 2, 3\}$  as well since no **req**-action is allowed and playing  $b, c$  from  $\{1, 2, 3\}$  has a positive probability to lead to  $\{4\}$ . Clearly,  $\sigma_0$  can be chosen positional, and on the example it is given by  $\sigma_0(\{2\}) = b$  and  $\sigma_0(\{3\}) = c$ .





Now, the canonical strategy  $\sigma_{can}^n$  is improved by letting  $\sigma_{can}^n(B) = \sigma_0(B)$  if  $B \in S_0 \setminus W_0$ , and leaving it unchanged otherwise. Under this new definition,  $\sigma_{can}^n$  is still almost-surely reaching **Goal**, and from any  $B \in S_0$ ,  $\sigma_{can}^n$  never proposes a **req**-action anymore:

► **Proposition 3.** Assuming  $s_0 \in \text{Win}$ , then: (i)  $\sigma_{can}^n$  is almost-surely winning, and (ii) for every  $\sigma_{can}^n$ -path  $\rho = \rho_1\rho_2$  with  $B(\rho_1) \in S_0$ ,  $\rho_2$  contains no **req**.

Given a strategy  $\sigma$ , we say that  $B$  is a  $\sigma$ -belief state if there exists a  $\sigma$ -path  $\rho$  with  $B(\rho) = B$ . We prove that  $S_0$  is optimal, in the following meaning:

► **Proposition 4.** Let  $\sigma$  be a strategy reaching **Goal** almost-surely from  $s_0$ , and  $B \notin S_0$  a  $\sigma$ -belief state. Then there exists a  $\sigma$ -path  $\rho = \rho_1\rho_2$  such that  $B(\rho_1) = B$  and  $\rho_2$  contains a **req**.

**Proof.** Let  $\sigma$  be an almost-surely winning strategy, and  $B$  a  $\sigma$ -belief state. Assume by contradiction that for every  $\sigma$ -path  $\rho_1\rho_2$  with  $B(\rho_1) = B$ ,  $\rho_2$  contains no **req**-action, and let us prove that  $B \in S_0$ . We design a strategy  $\sigma'$  in the MDP  $G_0$  from  $B$  as follows. For each run  $r = Ba_1B_1 \cdots a_nB_n$  in  $G_0$ , let  $\rho_r = Oa_1O_1 \cdots a_nO_n$  be the associated possible path in  $G$  with  $B_i \subseteq O_i$  for all  $i$ . The choice of  $O_i$  is unique, and it is always a part of **Part** since  $a_i \neq \text{req}$ . We let  $\sigma'(r) = \sigma(\rho_1\rho_r)$ . Now, it is easy to see that  $\sigma'$  reaches **Goal** almost surely from  $B$ , since  $\rho_1$  is a finite  $\sigma$ -path and  $\sigma$  reaches **Goal** with probability 1. As a consequence,  $B \in S_0$ . ◀

We can now define the set  $S_1$  of belief states for which at most one **req**-action is needed to reach **Goal** almost-surely. We let  $L_1$  be the set of belief states  $B$  such that for all  $s \in B$ ,  $\{s\} \in S_0$ . Playing a request from a state in  $L_1$  obviously leads to some state in  $S_0$ , from which winning without request is possible. Clearly,  $L_1 \cup S_0 \subseteq S_1$ . In fact,  $S_1$  is the set of all belief states from which there is a strategy to reach  $L_1 \cup S_0$  in  $G_0$  with probability one.  $S_1$  can be computed as follows:

1. Initialize a set  $X$  of belief states at  $\mathcal{B} \setminus \text{Lose}$ ,
2. Compute the set  $Y_X \subseteq \mathcal{B}$  of belief states which can reach  $L_1 \cup S_0$  while staying in  $X$ , using a smallest fixed point:
  - a. initialize  $Y_X$  at  $L_1 \cup S_0$ ,
  - b. add to  $Y_X$  all  $B \in X$  such that there exists  $a \in \text{Act}$  with  $\text{Supp}(\delta_0(B, a)) \subseteq X$  and  $\text{Supp}(\delta_0(B, a)) \cap Y_X \neq \emptyset$ .
3. If  $X \neq Y_X$  then set  $X := Y_X$  and goto step 2 again, else set  $S_1 = Y_X$  and quit.

The so-computed set  $S_1$  of belief states is the largest one such that from all belief states of  $S_1$  there is a strategy which allows to reach  $L_1 \cup S_0$  and which ensures to stay in  $S_1$ . Now, we improve once again the canonical strategy  $\sigma_{can}^n$  with,

- if  $B \in L_1 \setminus S_0$ ,  $\sigma_{can}^n(B) = \text{req}$ ,
- if  $B \in S_1 \setminus (L_1 \cup S_0)$ ,  $\text{Supp}(\sigma_{can}^n(B))$  is the uniform distribution over the set of actions  $a \in \text{Act}$  such that  $\text{Supp}(\delta_0(B, a)) \subseteq S_1$ ,
- otherwise, it is unchanged.

Notice that from  $S_1 \setminus (L_1 \cup S_0)$ , always at least one action  $a \in \text{Act}$  satisfies  $\delta_0(B, a) \subseteq S_1$ .

After this modification,  $\sigma_{can}^n$  still reaches **Goal** with probability 1 and from any  $B \in S_1 \setminus S_0$ ,  $\sigma_{can}^n$  proposes at most one **req**-action:

► **Proposition 5.** Assuming  $s_0 \in \text{Win}$ , then: (i)  $\sigma_{can}^n$  is almost-surely winning, and (ii) for every  $\sigma_{can}^n$ -path  $\rho = \rho_1\rho_2$  with  $B(\rho_1) \in S_1$ ,  $\rho_2$  contains at most one **req**.



**Proof.** (ii) is fairly easy to establish. Let  $\rho = \rho_1\rho_2$  be a  $\sigma_{can}^n$ -path with  $B = B(\rho_1) \in S_1$ . As  $\rho_2 = B \xrightarrow{a_1} B_1\rho_2'$  is a  $\sigma_{can}^n$ -path, the first action  $a_1$  ensures to stay in  $S_1$ : more precisely, if  $B \in L_1$ , then  $B_1 \in S_0$ , if  $B \in S_0$ , then  $B_1 \in S_0$ , and if  $B \in S_1 \setminus (L_1 \cup S_0)$  then  $B_1 \in S_1$ . Iterating this argument, beliefs in  $\rho_2$  always belong to  $S_1$ . In case  $L_1$  is never reached, request are never performed. Otherwise, as soon as it reaches  $L_1$ , a request is played, and the next belief state is in  $S_0$  from which no request are proposed anymore, according to the Proposition 3. Overall, (ii) is verified.

We now prove that from  $S_1$ ,  $L_1 \cup S_0$  is reached with probability 1, which proves that  $\sigma_{can}^n$  reaches **Goal** with probability 1 thanks to Proposition 3. The only paths from  $S_1$  which does not reach  $L_1 \cup S_0$  are those staying forever in  $S_1 \setminus (L_1 \cup S_0)$ . We now prove that from every state in  $S_1$ , there is a  $\sigma_{can}^n$ -path reaching  $L_1 \cup S_0$ , hence, a positive probability to reach  $L_1 \cup S_0$ . Together, these facts show that almost-surely  $L_1 \cup S_0$  will be reached from  $S_1$ . Assume now by contradiction that there is a state  $B \in S_1$  with  $B(\rho_1) = B$  and such that for all  $\sigma_{can}^n$ -path  $\rho = \rho_1\rho_2$ , we have  $B(\rho) \notin L_1 \cup S_0$ . Hence these paths  $\rho_2$  make no request. But this set of  $\rho_2$  paths cover exactly the set of path from  $B$  which stay within  $S_1$ . Considering the last iteration of the construction of  $S_1$ . As it is the last iteration,  $Y_X = X = S_1$  and  $B \in X$ . Now, in the construction of  $Y_X$ ,  $B$  can never be added to  $Y_X$ . Hence  $Y_X \neq X$ , a contradiction.  $\blacktriangleleft$

We prove that  $S_1$  is optimal, in the following sense:

► **Proposition 6.** Let  $\sigma$  be a strategy reaching **Goal** almost-surely from  $s_0$  and  $B \notin S_1$  a  $\sigma$ -belief state. Then there is a  $\sigma$ -path  $\rho = \rho_1\rho_2$  with  $B(\rho_1) = B$  and  $\rho_2$  contains at least two req-actions.

We can easily construct in this way by induction the sets  $(S_k)_{k \in \mathbb{N}}$  of belief states requiring at most  $k$  requests, until stabilization:  $S_{K+1} = S_K$ , which happens at worse for  $K = |\mathcal{B}|$ . Computing each  $S_i$  takes  $O(|\mathcal{B}|^2 \times |\text{Act}|)$ ; overall, the procedure is in time  $O(|\mathcal{B}|^3 \times |\text{Act}|)$ .

This easily improves the strategy  $\sigma_{can}^n$  by allowing requests only when they are needed. Then, denoting  $S_\infty = 2^Q \setminus S_K$ , we have the following optimality result:

► **Proposition 7.** For every strategy  $\sigma$  reaching goal almost surely and every  $\sigma$ -belief states  $B \in S_\infty$ , for all  $N \in \mathbb{N}$ , there exists a  $\sigma$ -path  $\rho = \rho_1\rho_2$  with  $B(\rho_1) = B$  and  $\rho_2$  contains at least  $N$  req-actions.

In our example, notice that  $S_1 = S_0$ , and hence  $S_\infty = \{\{1\}, \{1, 2, 3\}\}$ . Hence, the improved canonical strategy in our running example is defined by:  $\sigma_{can}^n(1) = a$ ,  $\sigma_{can}^n(2) = b$ ,  $\sigma_{can}^n(3) = c$  and  $\sigma_{can}^n(\{1, 2, 3\})$  assigns probability  $(n-1)/n$  to action  $a$  and probability  $1/n$  to req. We show now that the family of canonical strategies is, on this particular example, also optimal for average cost! The proof is educational to understand that even with a fixed strategy, computing the average cost in a POMDP is not easy as the set of possible stochastic belief states (precise distributions over states in the discrete belief states) is potentially infinite.

Let  $n \in \mathbb{N}$  and consider strategy  $\sigma_{can}^n$ . The game starts in state 1, and  $\sigma_{can}^n$  first decision is to perform an  $a$ . After this action, the discrete belief state is  $\{1, 2, 3\}$  and the probability to be in state 1 is  $1/3$ . In the sequel, we denote by  $E_k$  the expected number of requests following  $\sigma_{can}^n$  from  $\{1, 2, 3\}$  with probability  $1/3^k$  to be in state 1. Thanks to the observation above, the expected number of requests for  $\sigma_{can}^n$  is exactly  $E_1$ .

Assume now that the current belief state is  $\{1, 2, 3\}$  and the probability to be in state 1 is  $1/3^k$ . In this state, with probability  $1/n$ , a request is performed, which discloses state 1 with probability  $1/3^k$ ; with probability  $(n-1)/n$ ,  $a$  is played, and the resulting state is  $\{1, 2, 3\}$  with probability  $1/3^{k+1}$  to be in state 1. As a consequence,  $E_k = \frac{1}{n}(1 + \frac{1}{3^k}E_1) + \frac{n-1}{n}E_{k+1}$ .





From there, we derive:  $E_1 = 1 + \frac{1}{2n}$ . Hence, the average number of **req** asked by  $\sigma_{can}^n$  is smaller than  $1 + \frac{1}{2n}$ , for all  $n \in \mathbb{N}$ . In fact, we can prove that this family of strategy is optimal in the sense that no almost surely winning strategy can achieve an average number of request of 1 or less on this particular example.

#### 4 Hardness and undecidability results for the average cost

We turn now to a more general analysis of the strategies minimizing the average cost. Unfortunately, as we hinted before, this question is very hard to tackle. We show first that the problem of the existence of a strategy with cost smaller than a fixed threshold is in general undecidable, and that it is undecidable to approximate the optimal average cost. Moreover, we give concrete approximation factor (with respect to the size of the POMDP) up to which no optimal strategy can be computed with polynomial time algorithms. This obviously contrasts with the rather efficient algorithms we design in the previous section, which run in time polynomial in  $|\mathcal{B}|$ .

For a run  $r$  of  $G$ , we denote by  $val(r)$  the number of **req** in  $r$ . Let  $\sigma$  be a strategy reaching *Goal* from  $s_0$  with probability 1. Then we denote by  $val(\sigma)$  the expected value of  $val(r)$ , over all the  $\sigma$ -runs. Notice that  $val(\sigma) < \infty$  since  $\sigma$  is almost-surely winning.

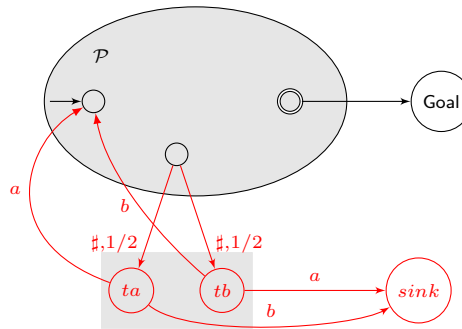
► **Definition 8.** The *value* of  $G$  is  $val(G) = \inf\{val(\sigma) \mid \sigma \text{ almost-surely winning}\}$ , where  $val(\sigma)$  denotes the expected number of requests under strategy  $\sigma$ .

We can now present our first negative result, namely that it is undecidable to compute the exact minimum average cost  $val(G)$ . This result should not be too surprising as optimizing a cost function in a POMDP is undecidable [13]. However, our result is stronger and harder to get, since our cost function is not arbitrary.

► **Theorem 9.** For all  $K > 0$ , it is undecidable to know whether  $val(G) \leq K$ .

**Proof.** Let  $\varepsilon \in (0, 1/2)$ . Take a Probabilistic Finite Automaton  $\mathcal{P}$  (a PFA for short, that is a POMDP with  $|\text{Part}| = 1$ ) such that either there exists a word accepted with probability at least  $1 - \varepsilon$  or all words are accepted with probability less than  $\varepsilon$ . It is undecidable to know which case holds [13]. From  $\mathcal{P}$ , we build a POMDP  $G$ , as illustrated on Figure 3 by adding four new states. This reduction ensures the following:  $\mathcal{P}$  accepts a word with probability greater than  $1 - \varepsilon$  if and only if  $val(G) < \frac{\varepsilon}{1-\varepsilon}$ . ◀

Actually, in the proof, we even show that if  $val(G) \geq \varepsilon/(1 - \varepsilon)$ , then  $val(G) \geq 1 - \varepsilon$ . That is,  $(\varepsilon/(1 - \varepsilon) + 1 - \varepsilon)/2$  is the best approximation one can make for this family of POMDP.



■ **Figure 3** Reduction to a variant of the emptiness problem for PFA

The approximation factor is thus  $\delta = (1/(1 - \varepsilon) - \varepsilon)(1 - \varepsilon)/2\varepsilon = (1 - \varepsilon(1 - \varepsilon))/2\varepsilon$ , which converges to infinity as  $\varepsilon$  converges to 0. As a consequence:

► **Corollary 10.** *For any  $\delta$ , it is undecidable to approximate  $\text{val}(G)$  with factor  $\delta$ .*

Notice however that for bigger  $\delta$ , the non-approximation result uses bigger and bigger PFA (and thus POMDP). The following result establishes the relationship between the number of states of the POMDP and the non-approximation factor.

► **Theorem 11.** *Assuming that  $P \neq NP$ , for any polynomial time algorithm  $\mathcal{A}$ , there exists a POMDP  $G$  with at most  $3n^2 + n$  states and at most  $9n^2 + 8n$  reachable belief states, such that  $\mathcal{A}$  computes a strategy with value  $\text{val}$  on  $G$  with:*

**approximation factor:**  $|\text{val} - \text{val}(G)|/\text{val}(G) \geq 2^{n-1}/n^2 - 1$ , and

**absolute approximation error:**  $|\text{val} - \text{val}(G)| \geq (2^{n-1}/n - n - 2)$ .

**Proof.** The proof is by reduction from 3-SAT. Let  $\varphi$  be a Boolean formula in conjunctive normal form with  $m$  clauses  $C_1 \cdots C_m$  over  $k$  variables  $x_1, \dots, x_k$ . We let  $n = k \times m$ . We also let  $\varepsilon = 1/2^n$ . From  $\varphi$ , we derive a POMDP  $G$  such that  $\varphi$  is satisfiable if and only if  $\text{val}(G) \leq 3n\varepsilon$ . We recall that  $\varphi$  is satisfiable if and only if for every clause  $C_i$ , one can choose one literal  $\ell_i$  among the three literals of  $C_i$  such that for all  $i, j \leq m$ , the choices of  $\ell_i$  and  $\ell_j$  do not conflict.

The actions that can be played are any of the  $2k$  literals  $\{x_i, \bar{x}_i \mid i \leq k\}$ , plus a dummy action. The POMDP starts in the initial state `init`, where only the dummy action can be fired, and with equal probability a variable  $x_i, i \leq k$  is chosen leading to state labeled  $(C_1, i)$ . Intuitively, the POMDP will remember actions played concerning this variable  $x_i$  and no other. All states  $(C_1, 1), \dots, (C_1, k)$  belong to the same part of the partition and thus the player does not know which variable is monitored.

From  $(C_1, i)$ , three actions are enabled, corresponding to the three literals in clause  $C_1$ . If the action played is  $x_j, \bar{x}_j$  with  $j \neq i$ , then the next state is  $(C_2, i)$  with probability  $1 - \varepsilon$ . With probability  $\varepsilon/2$  it is  $(C_2, x_i)$  and with probability  $\varepsilon/2$  it is  $(C_2, \bar{x}_i)$ . Intuitively, with small probability, the POMDP remembers wrongly that literal  $x_i$  or  $\bar{x}_i$  has been chosen. If the action played is  $x_i$ , then the next state is  $(C_2, x_i)$  with probability  $1 - \varepsilon$ ,  $(C_2, i)$  with probability  $\varepsilon/2$  and  $(C_2, \bar{x}_i)$  with probability  $\varepsilon/2$ . At last, if the action played is  $\bar{x}_i$ , then the next state is  $(C_2, \bar{x}_i)$  with probability  $1 - \varepsilon$ ,  $(C_2, i)$  with probability  $\varepsilon/2$  and  $(C_2, x_i)$  with probability  $\varepsilon/2$ . Transitions from  $(C_\ell, i)$  with  $\ell < m$  and  $i \leq k$  follow the same pattern.

Now, assume that the state is  $(C_\ell, \bar{x}_i)$ , that is the POMDP recalls (possibly wrongly) that  $\bar{x}_i$  has been played before. If the action played is  $x_i$ , then there is a conflict and the next state is `test`, which is the gadget, similar to the one in the proof of Theorem 9 that forces the player to perform a `req`-action in order not to lose, and then it reaches state `ok`. If the action played is any other, then the next state is  $(C_{\ell+1}, \bar{x}_i)$  with probability  $1 - \varepsilon$ , and  $(C_{\ell+1}, x_i)$  or  $(C_{\ell+1}, i)$  with probability  $\varepsilon/2$  each. From state  $(C_\ell, x_i)$ , the transitions are symmetric. At last, for the last clause, from  $(C_m, x_i)$ , all the actions lead to `ok` except action  $\bar{x}_i$  which leads to the test gadget.

This reduction ensures: there exists a strategy  $\sigma$  reaching `ok` with probability 1 and such that  $\text{val}(\sigma) \leq n\varepsilon$  if and only if the formula  $\varphi$  is satisfiable. Assume that there is a non conflicting choice of literals for every clause. Consider the strategy  $\sigma$  which chooses to play accordingly to this choice of literal, and when in the test gadget performs a `req`. With probability higher than  $1 - n\varepsilon$ , the POMDP remembers accurately the choice of literal  $x_i$  by  $\sigma$  at each step (there are less than  $n$  steps). As this choice is not conflicting, under this hypothesis, no `req` is played and `ok` is reached. Now, with probability less than  $n\varepsilon$ ,



the memory can be wrong at some point, and the worse case is to reach the **test**, in which case a unique requests is made before reaching **ok**. Thus,  $val(\sigma) \leq n\varepsilon$ . For the reverse implication, observe that under any strategy  $\sigma$  reaching **ok** with probability 1, there is at least one conflicting variable (successive choices of literal  $x_i$  and  $\bar{x}_i$ ). As in the reduction of Theorem 9 we assume without loss of generality that  $\sigma$  does not propose any **req** but in **test**. With probability at least  $1/n$ , the POMDP remembers the conflicting variable, and with probability at least  $(1 - n\varepsilon)$  the POMDP remembers accurately the first literal of  $x_i$  played, and then when  $\bar{x}_i$  is played, the POMDP goes to **test** where a **req** is played. Overall,  $val(\sigma) \geq (1/n - \varepsilon)$ .

As SAT is *NP*-complete and assuming that  $P \neq NP$ , no polynomial time algorithm can decide whether  $val(G) \geq (1/n - \varepsilon)$  or  $val(G) \leq n\varepsilon$ . To keep the error factor as low as possible, the safest is thus to play the average value, which proves the first item.

One can however notice that even if the approximation factor is large, the absolute gap between an approximation and the real value is rather small (less than 1), which would not be a huge concern in practice. We explain now how to keep the same factor while widening the absolute gap using a small trick. The previous reduction is enriched as follows: from state **ok** only the dummy action can be played, leading with probability  $\varepsilon$  to **Goal**, and with probability  $1 - \varepsilon$  back to **init**. The probability to reach **Goal** after seeing **init** exactly  $i$  times is  $\varepsilon(1 - \varepsilon)^i$ , that is the expected number of times **init** is seen is  $\sum_i i \cdot \varepsilon \cdot (1 - \varepsilon)^i \in [2^n - 2, 2^n]$  for  $n$  large enough. That is, if there is a non conflicting choice of literals, then  $val(G) \leq n/2^n \cdot 2^n = n$ . On the other hand, if all choice of 1 literal per clause is conflicting, then  $val(G) \geq (1/n - \varepsilon) \cdot (2^n - 2) \geq 2^n/n - 2$ . This concludes the proof. ◀

Notice that the first item of our result holds for POMDP without loops, that is in particular for finite horizon POMDP (horizon  $\leq n$ ). Compared with other results on non approximability of optimal cost in (finite horizon) POMDP [12], our reduction does not rely on the (at worse exponentially many) discrete belief states to encode the problem, but uses the actual probability to be in a state. Indeed, the family of graph considered has a polynomial number of reachable belief states (for which the algorithm of the previous section are efficient). Actually, if we relax the constraint of a polynomial number of beliefs, the proof can easily be simplified to obtain an infinite non approximability factor, since  $val(G) = 0$  if and only if the 3-SAT formula is satisfiable. It also proves that reasoning on beliefs is in some sense mandatory.

## 5 Conclusion

In this paper we investigated the problem of minimizing requests for full information in a POMDP in order to achieve a reachability objective with probability 1. On the one hand, the optimal worst-case cost is in  $\mathbb{N} \cup \{\infty\}$  and can be computed in polynomial time in the number of discrete beliefs (that is, exponential time at worse), together with a finite memory strategy that guarantees this optimal cost. On the other hand, the optimal average cost is in  $\mathbb{R}_{\geq 0}$  and can neither be computed nor approximated, and we provide large error factors for which no polynomial-time algorithm can approximate the average optimal cost up to that factor. In practice, despite the non-approximability result, quite accurate values can be obtained for some POMDPs using heuristics [14]. Our model can be enriched to allow several intermediate information levels, encoded by successive refinements of the partition, and for which the techniques developed here will certainly be useful.



**Acknowledgements** We would like to thank the CNRS PEPS UMI INSIS AABS for the financial support without which this work would not have taken place, as well as the referees for pointing out very interesting related bibliography.

---

## References

---

- 1 C. Baier, N. Bertrand, and M. Grösser. On decision problems for probabilistic Büchi automata. In *FOSSACS '08*, volume 4962 of *LNCS*, pages 287–301. Springer, 2008.
- 2 N. Bertrand, B. Genest, and H. Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *LICS '09*, pages 319–328. IEEE, 2009.
- 3 F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic observers. *Fundamenta Informaticae*, 88(4):497–540, 2008.
- 4 R. Chadha, A. P. Sistla, and M. Viswanathan. Power of randomization in automata on infinite strings. In *CONCUR '09*, volume 5710 of *LNCS*, pages 229–243. Springer, 2009.
- 5 K. Chatterjee and L. Doyen. The Complexity of Partial-Observation Parity Games. In *LPAR '10 (Yogyakarta)*, LNCS 6397, pages 1–14, 2010.
- 6 K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In *MFCS '10*, volume 6281 of *LNCS*, pages 258–269. Springer, 2010.
- 7 K. Chatterjee and R. Majumdar. Minimum attention controller synthesis for omega-regular objectives. In *FORMATS'11*, volume 6919 of *LNCS*, pages 145–159. Springer, 2011.
- 8 K. Chatterjee, R. Majumdar, and T. A. Henzinger. Controller synthesis with budget constraints. In *HSCC'08*, volume 4981 of *LNCS*, pages 72–86. Springer, 2008.
- 9 A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruńczyk. Energy and Mean-Payoff Games with Imperfect Information. In *CSL*, LNCS 6247, pages 260–274, 2010.
- 10 J. Fearnley. Exponential Lower Bounds for Policy Iteration. In *ICALP '10*, volume 6199 of *LNCS*, pages 551–562. Springer, 2010.
- 11 W. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. *OR*, 39:162–175, 1991.
- 12 C. Lusena, J. Goldsmith, and M. Mundhenk. Nonapproximability results for partially observable Markov decision processes. *JAIR*, 14, 2001.
- 13 O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003.
- 14 K. Murphy. A Survey of POMDP Solution Techniques. Technical report, 2000.
- 15 D. Thorsley and D. Teneketzis. Active acquisition of information for diagnosis and supervisory control of discrete event systems. *Discrete Event Dynamic Systems*, 17(4):531–583, 2007.
- 16 M. Tracol. Recurrence and transience for finite probabilistic tables. *TCS*, 412(12-14):1154–1168, 2011.
- 17 M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS '85*, pages 327–338. IEEE, 1985.

